

You are the owner of a railway system between n cities, numbered by integers from 1 to n . Each train travels from the start station to the end station according to a very specific timetable (always on time), without stopping anywhere between. A departure timetable is available on each station. Unfortunately each timetable contains only direct connections. A passenger that wants to travel from city p to city q is not limited to direct connections — he or she can change trains. Each change takes zero time, but a passenger cannot change from one train to the other if it departs before the first one arrives. People would like to have a timetable of all optimal connections. A connection departing from city p at A o'clock and arriving in city q at B o'clock is called optimal if there is no connection that begins in p not sooner than at A and ends in q not later than at B . We are only interested in connections that can be completed during the same day.

Write a program that:

- reads the number of cities n and timetables from the standard input,
- creates a timetable of optimal connections from city 1 to city n ,
- writes the answer to the standard output.

Input

The first line of the input contains an integer n ($2 \leq n \leq 100000$). The following lines contain n timetables for cities $1, 2, \dots, n$ respectively.

The first line of the timetable description contains only one integer m . Each of the following m lines corresponds to one position in the timetable and contains: departure time A , arrival time B ($A < B$) and destination city number t ($1 \leq t \leq n$) separated by single spaces. Departure time A and arrival time B are written in format $hh:mm$, where hh are two digits representing full hours ($00 \leq hh \leq 23$) and mm are two digits representing minutes ($00 \leq mm \leq 59$). Positions in the timetable are given in non-decreasing order according to the departure times. The number of all positions in all timetables does not exceed 1 000 000.

Output

The first line of the output contains an integer r — the number of positions in the timetable being the solution. Each of the following r lines contains a departure time A and an arrival time B separated by single space. The time format should be like in the input and positions in the timetable should be ordered increasingly according to the departure times. If there is more than one optimal connection with the same departure and arrival time, your program should output only one of them.

Sample Input

```
3
3
09:00 15:00 3
10:00 12:00 2
11:00 20:00 3
2
11:30 13:00 3
12:30 14:00 3
0
```

Sample Output

```
2
10:00 14:00
11:00 20:00
```